```
;==============================================================================
; CYGNUS BOOT 2.1
;==============================================================================
; Disassembled from binary version. I lost original source codes, but i
; reconstructed this with help of source codes for version 2.0.
; Between versions 2.0 and 2.1 are many differences, but some parts are same.
;
; I used this booter many years, and i know it is not ideal. With this source
; i will able repair bugs and maybe create new version with KMouse support etc...

        cpu     z80undoc

;==============================================================================
; Adresses of BASIC commands and parameters in line 10
;
; 10 RANDOMIZE USR VAL "23926": CLEAR VAL "65367": RANDOMIZE USR VAL "15616": REM : LOAD "a:        "

BASLN_RANDUSR   equ     23900                   ; last number in address RANDOMIZE USR VAL "15619"
BASLN_CLEAR     equ     23885                   ; first number in CLEAR VAL "65367"
BASLN_TRD_COMM  equ     23905                   ; address of TRDOS command (LOAD, RUN)
BASLN_TRD_DRV   equ     23907                   ; address of drive letter in TRDOS command
BASLN_TRD_NAME  equ     23909                   ; address of filename in TRDOS command
BASLN_TRD_CODE  equ     23918                   ; address of space for CODE in TRDOS command

TRD_BUFFER      equ     38921                   ; first 9 sectors from TRDOS disc = directory + system
sector
VRAM_BUFFER     equ     41225                   ; picture (VRAM) buffer
UNI_BUFFER      equ     48137                   ; universal buffer for all bigger data (max. 16kB)
                                                ; 65535 - 16384 - 1014 (space for User Defined Graphics
                                                (256B) and ...?)

BOOT_FONT       equ     34274                   ; bold font - same as other letters in user interface
                                                ; first character in font (space) + 256 bytes from this
                                                address
                                                ; this is address compatible with ROM

;==============================================================================

        org     32600

SCREEN          binclude        "grf/boot_2-1_gui_picture.bin"  ; GUI picture compressed by PRESSOR 5
BOOT_FONT_BIN   binclude        "grf/boot_2-1_font.bin"         ; bold font

;==============================================================================
; 35298 - start
        org     35298

START           push    hl                      ; store all registers
                push    de
                push    bc
                push    iy
                push    ix
                exx
                push    hl
                push    de
                push    bc
                exx
                push    af

                di
                ld      a,(23798)               ; get drive number, store
                ld      (23833),a
                and     3                       ; drive number can be {0,1,2,3}
                add     a,97                    ; convert to drive letter {a,b,c,d}
                ld      (BASLN_TRD_DRV),a
                call    CLS                     ; clear screen
                ld      a,7                     ; white BORDER
                out     (254),a
                ld      a,155                   ; init. 8255, all ports as input, needed for AMouse or
                Kempston Joystick
                out     (127),a
BOOT_RESTART    ld      hl,BOOT_FONT            ; set BASIC font, why? why this address?
```

```
                ld      (23606),hl
                ld      a,2                     ; set output channel 2
                call    5633
                call    TEXTOUT_2
                db      17,8,17,8,18,8
                db      19,8,20,0,21,0
                db      22,0,128
BETALOAD        call    SCREEN                  ; 32600 - display PRESSORed SCREEN = BOOT GUI
                jp      TRD_READ_DIR            ; 36725 - read whole TRDOS directory

;-------------------------------------------------------------------------------
; 35371 - file filter and file listing in 3 columns

FILETYPE_SWITCH call    FILE_TABLE              ; create table of pointers to TRDOS directory
                ld      a,3                     ; set default file type (BASIC)
                ld      (FILE_TYPE),a
                xor     a                       ; reset column number
                ld      (FILE_TYPE_TMP),a
FILETYPE_NEXT   ld      a,(FILE_TYPE)           ; 35383
                cp      3
                jp      nz,FILETYPE_NEXT_2      ; 35392
                xor     a
FILETYPE_NEXT_2 inc     a
                ld      (FILE_TYPE),a
                ld      bc,FTYPE_TABLE_L        ; 60 - search in 60 bytes number of file typ (1,2,3)
                ld      hl,FTYPE_TABLE          ; 35514 - start of table with file type names
                cpir
                ld      a,(hl)                  ; HL points to right table for this file type
                ld      (BASLN_TRD_COMM),a      ; then modify BASIC line
                inc     hl
                ld      a,(hl)
                ld      (BASLN_TRD_CODE),a
                inc     hl
                ld      a,(hl)
                ld      (PRINT_FN_FILTER+1),a   ; set file filter value
                inc     hl
                ld      a,(hl)
                ld      (PRINT_FN_INDEX+2),a    ; change index that points to ext. (basic, bytes) or
                                                ; length in sectors (snapshots)
                inc     hl
                push    de
                push    bc
                ld      de,BASLN_CLEAR
                ld      bc,5
                ldir
                pop     bc
                pop     de
                dec     hl
                ld      (FTYPE_MSGADDR+1),hl    ; set pointer to file type name
                call    PRINT_FNAMES
                jp      z,FTYPE_NOFILE          ; Z = no file with this file type, try next
                call    SOUND_CLICK
                call    TEXTOUT_2
                db      16,8,17,8,19,8          ; position for messages or name of file type
                db      22,2,21+128
FTYPE_MSGADDR   ld      hl,FTYPE_TABLE+9        ; address will changed
                inc     hl
                call    TEXTOUT_1               ; print file type name
                jp      MOUSE_GUI               ; start mouse driver

FTYPE_NOFILE    ld      a,(FILE_TYPE_TMP)       ; try next file type
                inc     a
                ld      (FILE_TYPE_TMP),a
                cp      4
                jp      z,FTYPE_NOFILE_2        ; Z = all file types tested, nothing found
                ld      hl,(FTYPE_MSGADDR+1)    ; set pointer to file type name
                jp      FILETYPE_NEXT

FTYPE_NOFILE_2  call    TEXTOUT_2               ; disc empty
                db      16,8,17,8,19,8
                db      22,2,21
                db      "NO FILE !",' '+128
```

```
                jp      TRDOS_ERRGUI            ; 37058

; 35514

FTYPE_TABLE     db      1                       ; 1 = basics
                db      239,32                  ; LOAD, space
                db      "B",8                   ; filtered value and index in filter
                db      "65367"                 ; RAMTOP for CLEAR in BASIC line
                db      "BASIC    ",' '+128

                db      2                       ; 2 = snapshots
                db      236,175                 ; GOTO, CODE
                db      192,13                  ; filtered value and index in filter
                db      "65367"                 ; RAMTOP for CLEAR in BASIC line
                db      "SNAPSHOT ",' '+128

                db      3                       ; 3 = binary files
                db      247,175                 ; RUN, CODE
                db      "C",8                   ; filtered value and index in filter
                db      "24500"                 ; RAMTOP for CLEAR in BASIC line
                db      "BYTES    ",' '+128

FTYPE_TABLE_L   equ     $-FTYPE_TABLE

;------------------------------------------------------------------------------
; 35574 - initialize PRINT_FNAMES, new file list
; reprint filenames in GUI without reinitialization table of file pointers

PRINT_FNAMES_0  call    PAUSENK
                call    CLEAR_WINDOWS
                ld      hl,15360                ; set ROM font for RST 16 output
                ld      (23606),hl
                ld      hl,(PRINT_FN_TABP_1)    ; get pointer in table (may points to 1. - 3. page)
                ld      (PRINT_FN_TABP_2),hl
                jp      PRINT_FN_1

;------------------------------------------------------------------------------
; 35595 - print filenames in three columns
; print filenames in GUI from first

PRINT_FNAMES    call    CLEAR_WINDOWS
                ld      hl,15360                ; set ROM font for RST 16 output
                ld      (23606),hl
                ld      hl,HEAD_POINTERS        ; table of pointers to TRDOS file headers
                ld      (PRINT_FN_TABP_2),hl    ; initialize table of file pointers
PRINT_FN_1      ld      a,6                     ; initialize position for first filename in first column
                ld      (LISTF_PRNPOS+1),a
                ld      a,1
                ld      (LISTF_PRNPOS+2),a
                xor     a
                ld      (PRINT_FN_RETVAL+1),a   ; set return value (0)
PRINT_FN_2      ld      e,(hl)                  ; ld de,(hl) - get pointer to file header
                inc     hl
                ld      d,(hl)
                dec     hl
                ld      a,e                     ; DE = 0? - is it end of pointers table?
                or      d
                jp      z,PRINT_FN_7
                push    de                      ; ld    ix,de
                pop     ix
PRINT_FN_INDEX  ld      a,(ix+8)                ; this index will changed (+8 basic & bytes, +?
snapshot)
PRINT_FN_FILTER cp      'B'                     ; file extension or length in sector (192 sectors for
snapshots)
                jp      nz,PRINT_FN_4
                push    hl
                ld      hl,LISTF_PRNPOS         ; position of filename at screen
                call    TEXTOUT_1
                ld      h,d
                ld      l,e
                ld      b,8
                call    TEXTOUT_3
```

```
                ld      a,1
                ld      (PRINT_FN_RETVAL+1),a   ; set return value (1)
                pop     hl
                ld      a,(LISTF_PRNPOS+1)
                inc     a
                cp      22
                jp      z,PRINT_FN_5
PRINT_FN_3      ld      (LISTF_PRNPOS+1),a
PRINT_FN_4      inc     hl
                inc     hl
                jp      PRINT_FN_2

PRINT_FN_5      ld      a,(LISTF_PRNPOS+2)
                add     a,9
                cp      28
                jp      z,PRINT_FN_6
                ld      (LISTF_PRNPOS+2),a
                ld      a,6
                jp      PRINT_FN_3

PRINT_FN_6      inc     hl
                inc     hl
                jp      PRINT_FN_END_2

; end of table

PRINT_FN_7      ld      a,(LISTF_PRNPOS+1)
                cp      21                      ; bottom line?
                jp      nz,PRINT_FN_END         ; NZ = no
                ld      a,(LISTF_PRNPOS+2)
                cp      19                      ; 3. column?
                jp      nz,PRINT_FN_END
                jp      PRINT_FN_END_2

PRINT_FN_END    ld      hl,HEAD_POINTERS        ; table of pointers to TRDOS file headers
PRINT_FN_END_2  ld      (PRINT_FN_TABP_1),hl
                ld      hl,BOOT_FONT            ; set BASIC font, why? why this address?
                ld      (23606),hl
                call    TEXTOUT_2
                db      22,2,8+128
                ld      hl,TRD_BUFFER+2302      ; 41223 = 38921 + 2048 + 254            ; last character
of discname
                set     7,(hl)
                ld      hl,TRD_BUFFER+2293      ; 41214 = 38921 + 2048 + 245            ; disc name
                call    TEXTOUT_1
                call    TEXTOUT_2
                db      22,3,7+128
                ld      a,(41197)
                ld      h,0
                ld      l,a
                call    NUMOUT_8
                call    TEXTOUT_2
                db      22,3,17+128
                ld      a,(TRD_BUFFER+2292)     ; 41213 = 38921 + 2048 + 244            ; deleted files
                ld      h,0
                ld      l,a
                call    NUMOUT_8
                call    TEXTOUT_2
                db      22,4,15+128
                ld      hl,(TRD_BUFFER+2277)    ; 41198 = 38921 + 2048 + 229            ; free sectors
                call    NUMOUT_16
PRINT_FN_RETVAL ld      a,1                     ; return value
                or      a
                ret

;-------------------------------------------------------------------------------
; 35798 - error messages

ERROR_MESSAGE   call    CLEAR_WINDOWS           ; clear windows for filenames
                call    TEXTOUT_2               ; set printing position
                db      19,8,16,8
                db      18,8,17,8
```

```
                db      22,2,21+128
                ld      a,(23823)               ; get TRDOS error code
                ld      hl,ERROR_MSG_TAB
                ld      bc,ERROR_MSG_LEN
                cpir                            ; search error message
                call    TEXTOUT_1               ; print it
                ret

; 35830

ERROR_MSG_TAB   db      3
                db      "DISK FULL",'!'+128
                db      6
                db      "NO DISK! ",' '+128

ERROR_MSG_LEN   equ     $-ERROR_MSG_TAB

; 35852
                db      "DISK ERRO",'R'+128     ; no used?

;-------------------------------------------------------------------------
; 35862 - print string with bit 7 set in last character from HL

TEXTOUT_1       ld      a,(hl)
                or      a
                ret     z
                and     127
                rst     16
                bit     7,(hl)                  ; test was it last character?
                inc     hl
                jp      z,TEXTOUT_1             ; Z = last character was not printed, repeat
                ret
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 35875 - print string with bit 7 set in last character, from address in stack

TEXTOUT_2       pop     hl
TEXTOUT_2_LOOP  ld      a,(hl)
                and     127
                push    de
                push    bc
                rst     16
                pop     bc
                pop     de
                bit     7,(hl)                  ; test was it last character?
                inc     hl
                jp      z,TEXTOUT_2_LOOP        ; Z = last character was not printed, repeat
                jp      (hl)

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 35891 - print B characters from address in HL

TEXTOUT_3       ld      a,(hl)
                or      a
                ret     z
                and     127
                push    bc
                rst     16
                pop     bc
                inc     hl
                djnz    TEXTOUT_3
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 35903 - convert numbers to digits and print it

NUMOUT_8        ld      c,' '                   ; call this for 8bits numbers
                jp      NUMOUT_3DIGITS

; 35908

NUMOUT_16       ld      c,' '                   ; call this for 16bits bunbers
                ld      de,10000
```

```
                call    NUMOUT_2
                ld      de,1000
                call    NUMOUT_2
NUMOUT_3DIGITS  ld      de,100
                call    NUMOUT_2
                ld      e,10
                call    NUMOUT_2
                ld      e,1
                ld      c,'0'

NUMOUT_2        ld      a,'0'-1
NUMOUT_3        inc     a
                or      a
                sbc     hl,de
                jp      nc,NUMOUT_3
                add     hl,de
                cp      '0'
                jp      nz,NUMOUT_5
                ld      a,c
NUMOUT_4        rst     16
                ret

NUMOUT_5        ld      c,'0'
                jp      NUMOUT_4

;------------------------------------------------------------------------------
; 35960 - clear window - clear space for filenames

CLEAR_WINDOWS   ld      b,128
                ld      hl,16577
                ld      de,16578
                call    CLEAR_WIN_1             ; 36028
                ld      b,128
                ld      hl,16586
                ld      de,16587
                call    CLEAR_WIN_1             ; 36028
                ld      b,128
                ld      hl,16595
                ld      de,16596
                call    CLEAR_WIN_1             ; 36028
                ld      hl,(FNAME_HLIGHT+1)     ; 36291+1
                ld      (JMENO_P2+1),hl         ; 36312+1
                ld      hl,256
                ld      (FNAME_HLIGHT+1),hl     ; 36291+1
                ld      hl,JMENO_P2             ; 36312
                call    TEXTOUT_1
                call    TEXTOUT_2
                db      17,8,21,0
                db      22,2,30
                db      16,8,18,8
                db      19,8+128
                ret

; 36028

CLEAR_WIN_1     push    bc                      ; clear one rectangle / column for filenames
                ld      bc,7                    ; 8 bytes
                ld      (hl),0
                push    hl
                push    de
                ldir                            ; fill with 0
                pop     de
                pop     hl
                call    DOWNHL                  ; next pixel line
                ex      de,hl
                call    DOWNHL
                ex      de,hl
                pop     bc
                djnz    CLEAR_WIN_1
                ret

;------------------------------------------------------------------------------
```

```
; 36052 - creates table of sorted pointers to files without deleted and "boot    B"

FILE_TABLE      ld      ix,TRD_BUFFER           ; pointer to first item in directory
                ld      a,(TRD_BUFFER+2276)     ; 41197 = 38921 + 2048 + 228 = number of files (all)
                ld      b,a
                ld      c,0                     ; file counter
                ld      de,16                   ; length of header in directory
                ld      hl,HEAD_POINTERS        ; table of pointers to TRDOS file headers
FILE_TAB_1      jp      FILE_TAB_BOOT

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 36071 - filter deleted files from table

FILE_TAB_2      ld      a,(ix+0)                ; get first byte from filename
                cp      1
                jp      z,FILE_TAB_LEAVE        ; Z = 1 = deleted file
                or      a
                jp      z,FILE_TAB_LEAVE        ; Z = 0 = deleted file
                ld      a,(ix+7)
                cp      127                     ; hidden file? is not standard TRDOS feature
                jp      z,FILE_TAB_LEAVE
                ld      a,ixl
                ld      (hl),a
                inc     hl
                ld      a,ixu
                ld      (hl),a
                inc     hl
FILE_TAB_LEAVE  add     ix,de                   ; compute address of next file in direcotory
                inc     c
                djnz    FILE_TAB_1
                ld      a,c
                ld      (HEAD_POINTERS_2),a     ; filtered but unsorted pointers? ... why writing
                counter here?
                ld      (hl),0
                inc     hl
                ld      (hl),0
                ld      b,c

;  BUBBLESORT

BBSORT          push    bc
                ld      ix,HEAD_POINTERS        ; table of pointers to TRDOS file headers
BBSORT_1        ld      l,(ix+0)                ; get one pointer to HL
                ld      h,(ix+1)
                ld      e,(ix+2)                ; get second pointer to DE
                ld      d,(ix+3)
                push    de                      ; store both pointers
                push    hl
                ld      a,d                     ; check second pointer - if is zero => end of table
                or      e
                jp      z,BBSORT_4              ; Z = next sorting cycle
BBSORT_2        ld      c,(hl)                  ; compare filenames ?
                ld      a,(de)
                cp      c
                inc     hl
                inc     de
                jp      z,BBSORT_2              ; repeat until both filenames are same
                pop     hl
                pop     de
                jp      nc,BBSORT_3             ; NC = first filenames are in right list
                ld      (ix+0),e                ; swap pointers
                ld      (ix+1),d
                ld      (ix+2),l
                ld      (ix+3),h
BBSORT_3        inc     ix
                inc     ix
                jp      BBSORT_1

BBSORT_4        pop     hl                      ; next sorting cycle
                pop     de
                pop     bc
                djnz    BBSORT
```

```
                ret                             ; done

;------------------------------------------------------------------------------
; 36176 - test, is this filename "boot    B"?
; IX points to start of tested filename

FILE_TAB_BOOT   push    ix
                push    hl
                push    bc
                ld      b,9                     ; length of filename with extension
                ld      hl,TRD_HEADER           ; TRDOS HEAD - boot
FILE_TAB_BOOT_1 ld      a,(ix+0)
                cp      (hl)
                inc     hl
                inc     ix
                jp      nz,FILE_TAB_BOOT_2
                djnz    FILE_TAB_BOOT_1
FILE_TAB_BOOT_2 pop     bc
                pop     hl
                pop     ix
                jp      nz,FILE_TAB_2           ; this item in directory is not "boot    B"
                jp      FILE_TAB_LEAVE

;------------------------------------------------------------------------------
; 36207 - print filenames or filename highlighting?

JMENO_L         ld      a,1
                jp      JMENO

JMENO_S         ld      a,10
                jp      JMENO

JMENO_P         ld      a,19
JMENO           ld      de,(FNAME_HLIGHT+1)
                ld      (FNAME_HLIGHT+2),a
                cp      d
                jp      nz,JMENO1
                ld      a,(MOUSE_Y+1)
                rra
                rra
                rra
                and     00011111b
                cp      e
                jp      nz,JMENO2
                jp      RET2BAS_LOAD            ; return to BASIC - LOAD file

; 36245

JMENO1          ld      a,(MOUSE_Y+1)           ; 37224+1
                rra
                rra
                rra
                and     00011111b
JMENO2          ld      (JMENO_P2+1),de         ; 36313
                ld      (FNAME_HLIGHT+1),a      ; 36291+1
                ld      hl,FNAME_HLIGHT         ; 36291
                call    TEXTOUT_1
                call    SOUND_EFFECT
                call    TEXTOUT_2
                db      17,8
                db      16,8,19,8,18,8
                db      22,2,30,21,0+128
                call    PAUSENK
                jp      MOUSE_GUI               ; start mouse driver

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 36291 - cursor/highlighting parameters

FNAME_HLIGHT    db      22,0,1,16,8             ; 36291 cursor position in one of filenames columns
                db      21,1,19,8,18,8
                db      17,4
                db      "        "              ; 8x space
```

```
JMENO_P2        db      22,7,1,16,8             ; 36312 deletes discname
                db      21,1,19,8,18,8
                db      17,6
                db      "       ",' '+128       ; 8x space

;-------------------------------------------------------------------------------
; 36333 - compute address of selected filename in table of names
; Z = file found - and file header copied at right place

FIND_SELFILE    ld      a,(PRINT_FN_FILTER+1)   ; set file filter value
                ld      (FIND_SF_FILTER+1),a
                ld      a,(PRINT_FN_INDEX+2)    ; change index that points to ext. (basic, bytes) or
                                                ; length in sectors (snapshots)
                ld      (FIND_SF_INDEX+2),a
                ld      hl,0
                ld      de,16                   ; length of TRDOS header
                ld      a,(FNAME_HLIGHT+2)      ; set X position for highlighting
                dec     a
                jp      z,FIND_SF_2
FIND_SF_1       add     hl,de
                sbc     a,9                     ; 1. 2. 3. column ?
                jp      nz,FIND_SF_1
FIND_SF_2       ld      a,(FNAME_HLIGHT+1)      ; filename highlighting - line
                sbc     a,5
                ld      e,a
                add     hl,de
                ld      e,l
                ld      d,h
                ld      hl,(PRINT_FN_TABP_2)
                dec     hl
                dec     hl
FIND_SF_NEXT    inc     hl
                inc     hl
                ld      a,(hl)
                ld      ixl,a
                inc     hl
                ld      a,(hl)
                ld      ixu,a
                dec     hl
                or      (hl)
                jp      z,FIND_SF_NFOUND        ; Z = file not found, user clicked at empty space in
                                                ; filenames listing
FIND_SF_INDEX   ld      a,(ix+8)                ; index - copy from file filter
FIND_SF_FILTER  cp      'C'                     ; file extension - copy from file filter
                jp      nz,FIND_SF_NEXT         ; NZ = next item (filter rule not match)
                dec     de
                ld      a,d
                or      e
                jp      nz,FIND_SF_NEXT         ; NZ = next item
                ld      e,(hl)
                inc     hl
                ld      d,(hl)
                ex      de,hl
                ld      de,BASLN_TRD_NAME       ; name in BASIC command
                ld      bc,8
                push    hl
                ldir
                pop     hl
                xor     a
                ret

FIND_SF_NFOUND  xor     a                       ; 0, NZ, NC
                cp      1                       ; set NZ, but A = 0
                ret

;-------------------------------------------------------------------------------
; 36426 - from mouse cursor coordinates search clicked icon
; call this if you want detect clicks on icons

MOUSE_GUI       call    MOUSE_DRIVER            ; call mouse driver, returns coordinates
MOUSE_XYICO     xor     a
```

```
                ld      hl,ICON_INDEX           ; pointer to number of icon in icons list
                ld      (hl),a
                ld      a,(MOUSE_X+1)           ; get coordinates to DE
                ld      d,a
                ld      a,(MOUSE_Y+1)
                ld      e,a
                ld      bc,4
                ld      ix,GUI_ICON_FOUND       ; icon found, jump to execution code
MOUSE_XYNEXTICO inc     (hl)
                add     ix,bc
                ld      a,(ix+0)
                cp      255
                jp      z,GUI_NO_ICON           ; no icon found
                ld      a,(ix+0)
                cp      d
                jp      nc,MOUSE_XYNEXTICO
                ld      a,(ix+1)
                cp      d
                jp      c,MOUSE_XYNEXTICO
                ld      a,(ix+2)
                cp      e
                jp      nc,MOUSE_XYNEXTICO
                ld      a,(ix+3)
                cp      e
                jp      c,MOUSE_XYNEXTICO
GUI_JP          jp      ICONS_NORMAL            ; any icon found, jump to code

GUI_NO_ICON     xor     a
GUI_ICON_FOUND  ld      (hl),a
                jp      MOUSE_GUI               ; start mouse driver

XYIKON          db      0,9,0,9                 ; quit
                db      224,248,48,56
                db      224,248,64,72
                db      224,248,80,88
                db      224,248,96,104
                db      168,208,32,40
                db      160,248,16,24
                db      224,248,160,176
                db      216,248,32,40
                db      8,72,48,175
                db      80,144,48,175
                db      152,216,48,175
                db      224,248,112,120
                db      224,248,128,136
                db      224,248,144,152
                db      255

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 36557 - jump to execution code for every icon

ICONS_NORMAL    xor     a
                ld      (FILE_TYPE_TMP),a
                ld      a,(ICON_INDEX)          ; number of selected icon
                cp      1
                jp      z,RET2BAS_QUIT          ; 38303
                cp      2
                jp      z,TRD_DRIVE_A           ; 36684
                cp      3
                jp      z,TRD_DRIVE_B           ; 36688
                cp      4
                jp      z,TRD_DRIVE_C           ; 36693
                cp      5
                jp      z,TRD_DRIVE_D           ; 36698
                cp      6
                call    z,RET2BAS_8255OUT       ; 38370
                cp      7
                jp      z,FILETYPE_NEXT         ; 35383
                cp      8
                call    z,PRINT_FNAMES_0        ; 35574
                cp      9
                jp      z,TRDOS_AUTOCOPY        ; 37070
```

```
                cp      10
                jp      z,JMENO_L               ; 36207
                cp      11
                jp      z,JMENO_S               ; 36212
                cp      12
                jp      z,JMENO_P               ; 36217
                cp      13
                jp      z,PICTURE_VIEW          ; 38084
                cp      14
                jp      z,ISOROM_INSTALL
                cp      15
                jp      z,FILE_INFO
                jp      MOUSE_GUI               ; start mouse driver

; 36642 - limited icons list for error situations - no disc, disc error ...

ICONS_ERROR     ld      a,(ICON_INDEX)          ; number of selected icon
                ld      hl,ICONS_NORMAL
                ld      (GUI_JP+1),hl
                cp      1
                jp      z,RET2BAS_QUIT          ; 38303
                cp      2
                jp      z,TRD_DRIVE_A           ; 36684
                cp      3
                jp      z,TRD_DRIVE_B           ; 36688
                cp      4
                jp      z,TRD_DRIVE_C           ; 36693
                cp      5
                jp      z,TRD_DRIVE_D           ; 36698
                cp      9                       ; why? this cannot work, but when error is disk empty
too?
                jp      z,TRDOS_AUTOCOPY        ; 37070
                jp      TRDOS_ERRMODE

;------------------------------------------------------------------------------
; 36684 - drive selector

TRD_DRIVE_A     xor     a
                jp      TRD_DRIVE_CMN

TRD_DRIVE_B     ld      a,1
                jp      TRD_DRIVE_CMN

TRD_DRIVE_C     ld      a,2
                jp      TRD_DRIVE_CMN

TRD_DRIVE_D     ld      a,3

TRD_DRIVE_CMN   ld      (TRDOS_SELDRIVE),a      ; store drive number
                ld      (23833),a               ; set TRDOS variables
                ld      (23798),a
                add     a,'a'                   ; convert drive number to letter a .. d
                ld      (BASLN_TRD_DRV),a
                ld      a,(TRDOS_SELDRIVE)      ; TYPDRIVE+1
                ld      c,1
                call    TRDOS                   ; select drive
                jp      BOOT_RESTART

;------------------------------------------------------------------------------
; 36725 - read whole TRDOS directory

TRD_READ_DIR    ld      bc,2309                 ; B = 5, C = 9
                ld      hl,TRD_BUFFER           ; directory buffer
                ld      de,0                    ; track = 0, sector = 0
                call    TRDOS
                jp      FILETYPE_SWITCH

;------------------------------------------------------------------------------
; 36740 - ISOROM loader and installer
; this code is buggy - doesn't work, when SRAM is empty
; or if page 0 not contains 48k BASIC
```

```
ISOROM_LOAD     ld      c,24                    ; TRDOS INIT - verify disk if it is TRDOS formated
                call    TRDOS
                ld      hl,UNI_BUFFER
                ld      a,(ix+13)               ; get file length/256
                cp      64
                jp      z,ISOROM_LD_1           ; only one 16kB page of ROM
                ld      d,(ix+15)               ; load first half of 32kB file
                ld      e,(ix+14)
                ld      b,64
                ld      c,5
                call    TRDOS
                xor     a                       ; ZX 128 ROM page 0
                out     (253),a
                inc     a                       ; ISOROM 128 SRAM with write enabled
                out     (239),a
                push    de
                push    hl
                push    bc
                ld      hl,UNI_BUFFER           ; copy ROM page to ISOROM 128 SRAM
                ld      de,0                    ; target
                ld      bc,16384                ; length
                ldir
                pop     bc
                pop     hl
                pop     de
                ld      a,16
                out     (253),a
                xor     a
                out     (239),a
                ld      a,d
                add     a,4
                ld      d,a
                jp      ISOROM_LD_2
; 36806

ISOROM_LD_1     ld      d,(ix+15)
                ld      e,(ix+14)
ISOROM_LD_2     ld      bc,16389                ; 64 * 256 + 5
                call    TRDOS
                ld      a,16                    ; ZX 128 ROM page 1
                out     (253),a
                ld      a,1                     ; ISOROM 128 SRAM with write enabled
                out     (239),a
                push    de
                push    hl
                push    bc
                ld      hl,UNI_BUFFER           ; copy ROM page to ISOROM 128 SRAM
                ld      de,0                    ; target
                ld      bc,16384                ; length
                ldir
                pop     bc
                pop     hl
                pop     de
                xor     a
                out     (239),a
                ld      c,a
                call    TRDOS
                jp      MOUSE_GUI               ; start mouse driver

;------------------------------------------------------------------------------
; 36853 - screen (6912/6144), pressored picture od anything else loader
; HL must points to TRDOS header of selected file

TRD_LOAD_PIC    push    hl
                pop     ix
                ld      c,24                    ; TRDOS INIT
                call    TRDOS
                ld      hl,UNI_BUFFER           ; target
                ld      c,5                     ; TRDOS service
                ld      a,(ix+13)               ; IX = ? points to TRDOS header?
                cp      27                      ; length in sectors => 6912 bytes, picture with colors
                jp      z,TRD_LOAD_PIC_1        ; Z = length is 27 sectors
```

```
                jp      c,TRD_LOAD_PIC_1        ; C = picture is not longer
                ld      a,27
                push    hl
                ld      hl,6912                 ; set file length in viewer
                ld      (PIC_VIEW_LEN+1),hl
                pop     hl
                jp      TRD_LOAD_PIC_2

TRD_LOAD_PIC_1  push    hl
                ld      l,(ix+11)               ; get real length is file smaller than 6912 bytes
                ld      h,(ix+12)
                ld      (PIC_VIEW_LEN+1),hl     ; set file length in viewer
                pop     hl
TRD_LOAD_PIC_2  ld      b,a
                ld      d,(ix+15)
                ld      e,(ix+14)
                call    TRDOS
                ld      c,0
                call    TRDOS
                call    PIC_VIEWER
                call    SCREEN_RESTORE
                jp      MOUSE_GUI               ; start mouse driver

;-------------------------------------------------------------------------------
; 36925 - TRDOS calling - without BASIC error messages

TRDOS           ld      (TRDOS_A+1),a
                ld      (TRDOS_DE+1),de
                ld      (TRDOS_BC+1),bc
                ld      (TRDOS_HL+1),hl
                xor     a
                push    hl
                ld      (23823),a
                ld      hl,(23613)
                ld      (TRDOS_2+1),hl
                push    de
                push    hl
                push    ix
                push    iy
                ld      hl,(MOUSE_DRIVER+1)
                ld      b,h
                ld      c,l
                ld      ix,MCURSOR_CLOCK        ; sprite address - clock = wait when beta working with
                                                floppy disc
                exx
                ld      hl,MCURSOR_BUFFER
                exx
                ld      a,b
                call    8881                    ; compute pixel address in VRAM
                ld      (MCURSOR_CLEAR+1),hl    ; set address for cursor clearing
                ld      (MCURSOR_DRAW_2+1),a    ; set offset in byte to righ
                call    MCURSOR_DRAW
                di
                ld      hl,TRDOS_1
                push    hl
                ld      (23613),sp
TRDOS_HL        ld      hl,TRD_BUFFER           ; directory buffer
TRDOS_BC        ld      bc,2309                 ; B = 5, C = 9
TRDOS_DE        ld      de,0                    ; track = 0, sector = 0
TRDOS_A         ld      a,120
                ld      ix,0
                ld      iy,23610
                exx
                ld      hl,0
                ld      de,0
                ld      bc,0
                exx
                di
                call    15635                   ; call TRDOS
                pop     hl
TRDOS_1         di
                ld      sp,(23613)
```

```
                call    MCURSOR_CLEAR
                pop     hl
                pop     iy
                pop     ix
                pop     hl
                pop     de
TRDOS_2         ld      hl,65364
                ld      (23613),hl
                pop     hl
                ld      a,(23823)
                or      a
                ret     z
                pop     af
TRDOS_ERRMODE   call    ERROR_MESSAGE           ; print error message
TRDOS_ERRGUI    ld      hl,ICONS_ERROR          ; redirect to simplified GUI (user can only select drive
and reload disc)
                ld      (GUI_JP+1),hl
                call    MOUSE_DRIVER            ; and call mouse driver, user must be able select other
                drive or reload disc
                jp      MOUSE_XYICO

;-------------------------------------------------------------------------------
; 37070 - save booter

TRDOS_AUTOCOPY  ld      a,(23823)               ; reset error code
                or      a
                jp      nz,TRDOS_ERRMODE
                ld      hl,13352
                ld      a,h
                ld      (MOUSE_Y+1),a
                ld      a,l
                ld      (MOUSE_X+1),a
                ld      (MOUSE_DRIVER+1),hl
                ld      a,255
                ld      (23610),a
                ld      hl,41194                ; 48137 - 6912 - 31 ?
                ld      a,(hl)                  ; set first sector for this file
                ld      (TRD_HEADER_SEC),a
                inc     hl
                ld      a,(hl)
                ld      (TRD_HEADER_TRCK),a     ; set first track for this file
                ld      a,9
                ld      (23761),a
                ld      hl,TRD_HEADER
                ld      bc,19                   ; B = 0, C = 19
                call    TRDOS
                ld      bc,12                   ; B = 0, C = 12
                call    TRDOS
                jp      BOOT_RESTART

;-------------------------------------------------------------------------------
; 37131 - play sound effect
; first is short noise click and second is longer noise

SOUND_CLICK     ld      hl,15616                ; ROM address
                ld      de,1000                 ; length
                jp      SOUND_EFFECT_1

SOUND_EFFECT    ld      hl,0                    ; ROM address
                ld      de,3500                 ; length
SOUND_EFFECT_1  ld      a,(hl)
                and     16                      ; speaker bit
                or      7                       ; BORDER 7
                out     (254),a                 ; play
                inc     hl
                dec     de                      ; repeat until DE > 0
                ld      a,d
                or      e
                jp      nz,SOUND_EFFECT_1
                ret

;-------------------------------------------------------------------------------
```

```
; 37161 - amouse and keyboard driver

MOUSE_DRIVER    ld      hl,13352                ; 40,52 - last coordinates
                push    hl
                ld      b,h                     ; HL > BC
                ld      c,l
                ld      ix,MCURSOR_ARROW        ; cursor sprite + mask (2x 15 pixels height)
                exx
                ld      hl,MCURSOR_BUFFER       ; buffer for background under cursor
                exx
                ld      a,b
                call    8881                    ; compute pixel address in VRAM
                ld      (MCURSOR_CLEAR+1),hl    ; set address for cursor clearing
                ld      (MCURSOR_DRAW_2+1),a    ; set offset in byte to righ
                call    MCURSOR_DRAW
                ld      a,(MOUSE_HID)
                or      a
                jp      nz,MKEY_DRIVER
                ld      bc,150*256              ; number of port 31 readings 150x, C = ?
MOUSE_AM_LOOP   push    bc
                ld      b,c
                in      a,(31)                  ; read Kempston Joystick port (AMouse)
                ld      de,MOUSE_DIRTAB_0       ; to DE last state od mouse optical sensors
                call    MOUSE_TTL_DIR           ; compute difference in X axis
MOUSE_X         ld      a,40                    ; 37209 - X coordinate
                sub     (hl)                    ; sub 0,1,255 from table
                jr      z,MOUSE_X_OUT           ; Z = X out of range, don't store
                ld      (MOUSE_X+1),a
MOUSE_X_OUT     inc     de
                in      a,(31)
                rrca
                call    MOUSE_TTL_DIR           ; compute difference in Y axis
MOUSE_Y         ld      a,52                    ; 37224 - Y coordinate
                sub     (hl)                    ; sub 0,1,255 from table
                jr      z,MOUSE_Y_OUT           ; Z = Y out of range, don't store
                cp      192                     ; bottom limit
                jr      nz,MOUSE_Y_STORE
                ld      a,191                   ; out of bottom border, repair value
MOUSE_Y_STORE   ld      (MOUSE_Y+1),a
MOUSE_Y_OUT     pop     bc
                djnz    MOUSE_AM_LOOP
                xor     a                       ; test keyboard - all keys
                in      a,(254)
                cpl
                and     00011111b
                jr      nz,MDRV_SET_KEYB        ; NZ = any key pressed
                ld      a,(MOUSE_X+1)           ; store coordinates
                ld      l,a
                ld      a,(MOUSE_Y+1)
                ld      h,a
                ld      (MOUSE_DRIVER+1),hl
                jp      MDRV_INT_50

MDRV_SET_KEYB   ld      a,1                     ; switch driver to keyboard
                ld      (MOUSE_HID),a
                jp      MDRV_INT_50

; 37271

MDRV_SET_AMOUS  xor     a                       ; switch to AMouse driver
                ld      (MOUSE_HID),a
                ld      hl,(MOUSE_DRIVER+1)     ; copy coordinates - init. mouse driver, it must
                starts/continue
                ld      a,l                     ; cursor moving from same place as keyboard driver
                ld      (MOUSE_X+1),a
                ld      a,h
                ld      (MOUSE_Y+1),a
                jp      MDRV_INT_50

; 37289 - keyboard driver

MKEY_DRIVER             push    hl
```

```
                push    bc
                call    MOUSE_KEYBOARD
                pop     bc
                ld      hl,(MOUSE_DRIVER+1)     ; get coordinates X = L, Y = H
                ld      c,3                     ; speed of movement
                bit     3,d                     ; test key up
                jp      z,MKEY_DOWN
                ld      a,h
                sub     c
                ld      h,a
                jp      nc,MKEY_DOWN
                ld      h,0
MKEY_DOWN       bit     2,d                     ; 37313 - test key down
                jp      z,MKEY_LEFT
                ld      a,h
                add     a,c
                ld      h,a
                cp      189
                jp      c,MKEY_LEFT
                ld      h,190
MKEY_LEFT       bit     1,d                     ; 37328 - test key left
                jr      z,MKEY_RIGHT
                ld      a,l
                sub     c
                ld      l,a
                jp      nc,MKEY_RIGHT
                ld      l,0
MKEY_RIGHT      bit     0,d                     ; 37340 - test key right
                jr      z,MKEY_COMMON
                ld      a,l
                add     a,c
                ld      l,a
                cp      252
                jp      c,MKEY_COMMON
                ld      l,252
MKEY_COMMON     ld      (MOUSE_DRIVER+1),hl     ; 37354 - store coordinates
                ld      a,d
                ld      (MDRV_KEYSTATE+1),a             ; store state of keys
                ld      a,l                     ; store coordinates again
                ld      (MOUSE_X+1),a
                ld      a,h
                ld      (MOUSE_Y+1),a
                pop     hl
                in      a,(31)                  ; test Kempston Joystik port
                bit     7,a
                jp      nz,MDRV_INT_50          ; NZ = bit 7 set => maybe interface is not connected
                and     3                       ; check lower 3 bits
                cp      3                       ; all 1 then it cannot be joystick, maybe AMouse or
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒nothing
                jp      z,MDRV_SET_AMOUS        ; switch to AMouse driver
MDRV_INT_50     ei
                halt
                di
                call    MCURSOR_CLEAR           ; delete cursor
                pop     hl
                in      a,(31)                  ; check if port 31 presents
                bit     7,a
                jp      nz,MDRV_KEYSTATE                ; NZ = no port 31, jump over Kempson J. or
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒AMouse fire button test
                and     01110000b               ; leave only bits for fire buttons
                jp      nz,MDRV_FIRE            ; NZ = any fire button pressed
MDRV_KEYSTATE   ld      a,0                     ; check stored keyboard state
                bit     4,a
                jp      nz,MDRV_FIRE            ; NZ = keyboard fire key pressed
                ld      bc,49150
                in      a,(c)
                and     1
                jp      z,MDRV_KEYS_ENTER
                ld      bc,63486
                in      a,(c)
                cpl
                and     15
```

```
                jp      nz,MDRV_KEYS_1234
                jp      MOUSE_DRIVER

MDRV_FIRE       exx                             ; finish mouse driver, fire pressed = click somewhere
                ld      hl,0
                exx
                ret

MDRV_KEYS_ENTER exx
                pop     hl
                ld      hl,0
                exx
                jp      BOOT_RESTART            ; restart booter?

MDRV_KEYS_1234  exx
                ld      hl,0
                exx
                ld      b,a
                bit     0,b
                ld      hl,13544                ; 256*52+232 - simulates click on right icon, set
                                                correct coordinates
                jp      nz,MDRV_KEYS1234_2
                bit     1,b
                ld      hl,17640                ; 256*68+232
                jp      nz,MDRV_KEYS1234_2
                bit     2,b
                ld      hl,21736                ; 256*84+232
                jp      nz,MDRV_KEYS1234_2
                ld      hl,25832                ; 256*100+232
MDRV_KEYS1234_2 ld      a,l
                ld      (MOUSE_X+1),a
                ld      a,h
                ld      (MOUSE_Y+1),a
                call    PAUSENK
                ret

MOUSE_TTL_DIR   and     5                       ; compute new movement direction for Amiga TTL mouse
                ld      c,a
                ld      a,(de)
                rlca
                and     10
                or      c
                ld      (de),a
                ld      c,a
                ld      hl,MOUSE_DIRTAB
                add     hl,bc
                ret

; 37509 - mouse driver variables

MOUSE_DIRTAB_0  db      0,0
MOUSE_HID       db      0                       ; 1 = keyboard or 0 = mouse - human interface devices

; 37512 - table for calculating the direction of movement AMouse

MOUSE_DIRTAB    db      0,1,255,0
                db      255,0,0,1
                db      1,0,0,255
                db      0,255,1,0

; mouse buffer and graphics

MCURSOR_BUFFER  db      00101000b,00111100b     ; 40,60         - cursor buffer
                db      00101000b,01000000b     ; 40,64
                db      00010000b,01111110b     ; 16,126
                db      00000000b,00000000b     ; 0,0
                db      00000000b,00000000b     ; 0,0
                db      00000000b,00010000b     ; 0,16
                db      00011100b,00010000b     ; 28,16
                db      00100000b,00010000b     ; 32,16
                db      00100000b,00010000b     ; 32,16
                db      00100000b,00010000b     ; 32,16
```

```
                db      00011100b,00001100b     ; 28,12
                db      00000000b,00000000b     ; 0,0
                db      00000000b,00000000b     ; 0,0
                db      00010000b,00000000b     ; 16,0
                db      00111000b,00111000b     ; 56,56

MCURSOR_ARROW   db      00000000b               ; 0             - cursor sprite / arrow
                db      01000000b               ; 64
                db      01100000b               ; 96
                db      01110000b               ; 112
                db      01111000b               ; 120
                db      01111100b               ; 124
                db      01111110b               ; 126
                db      01111000b               ; 120
                db      01001000b               ; 72
                db      00001000b               ; 8
                db      00000100b               ; 4
                db      00000100b               ; 4
                db      00000010b               ; 2
                db      00000010b               ; 2
                db      00000000b               ; 0

                db      11100000b               ; 224           - cursor mask / arrow
                db      11110000b               ; 240
                db      11111000b               ; 248
                db      11111100b               ; 252
                db      11111110b               ; 254
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111100b               ; 252
                db      11111110b               ; 254
                db      00011110b               ; 30
                db      00001111b               ; 15
                db      00001111b               ; 15
                db      00000111b               ; 7
                db      00000111b               ; 7

MCURSOR_CLOCK   db      00000000b               ; 0             - cursor sprite / clock
                db      11111111b               ; 255
                db      01000010b               ; 66
                db      01010110b               ; 86
                db      01101010b               ; 106
                db      00110100b               ; 52
                db      00111100b               ; 60
                db      00011000b               ; 24
                db      00100100b               ; 36
                db      00100100b               ; 36
                db      01000110b               ; 70
                db      01101010b               ; 106
                db      01110110b               ; 118
                db      11111111b               ; 255
                db      00000000b               ; 0

                db      11111111b               ; 255           - cursor mask / clock
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      01111110b               ; 126
                db      01111110b               ; 126
                db      00111100b               ; 60
                db      01111110b               ; 126
                db      01111110b               ; 126
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111111b               ; 255
                db      11111111b               ; 255

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37618 - draw mouse cursor
```

```
; IX    = graphics data (+15 mask)
; HL'   = old background
; HL    = address of top left byte in VRAM

MCURSOR_DRAW    ld      b,15                    ; 15 pixels height of cursor
MCURSOR_DRAW_1  push    bc                      ; first delete cursor old cursor
                push    hl
                push    hl
                ld      h,0
                ld      l,(ix+0)                ; get graphics to HL
                ld      d,h
                ld      e,(ix+15)               ; get mask to DE
                inc     ix                      ; next byte
                ld      a,8
MCURSOR_DRAW_2  sub     0                       ; 37636 + 1 argument in sub
                ld      b,a
MCURSOR_DRAW_3  add     hl,hl                   ; HL * 2
                ex      de,hl                   ; DE * 2
                add     hl,hl
                ex      de,hl
                djnz    MCURSOR_DRAW_3

                ex      (sp),hl
                pop     bc
                ld      a,(hl)
                exx
                ld      (hl),a
                inc     hl
                exx
                ld      a,d
                cpl
                and     (hl)
                or      b
                ld      (hl),a
                call    RIGHTL                  ; compute address of byte right from first
                ld      a,(hl)
                exx
                ld      (hl),a
                inc     hl
                exx
                ld      a,e
                cpl
                and     (hl)
                or      c
                ld      (hl),a
                pop     hl

                call    DOWNHL                  ; compute address of next line
                pop     bc                      ; pop number of remaining lines
                djnz    MCURSOR_DRAW_1          ; next line
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37678 - clear mouse cursor, draw stored backgroud

MCURSOR_CLEAR   ld      hl,17605                ; address where was drawed cursor
                ld      b,15                    ; 15 pixel lines
                ld      de,MCURSOR_BUFFER       ; address of stored background
MCURSOR_CLEAR_1 ld      a,(de)                  ; copy first byte from buffer to VRAM
                ld      (hl),a
                inc     de
                push    hl
                call    RIGHTL                  ; compute address of byte right from first
                ld      a,(de)                  ; copy second byte from buffer to VRAM
                ld      (hl),a
                inc     de
                pop     hl
                call    DOWNHL                  ; next pixel line
                djnz    MCURSOR_CLEAR_1         ; repeat
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
; 37703 - keyboard driver
; returns pressed keys in A

MOUSE_KEYBOARD  ld      hl,MOUSE_KEYTAB
                ld      de,5
MOUSE_KB_CLP    ld      c,(hl)                  ; get first byte of port address
                inc     hl
                ld      b,(hl)                  ; get second byte of port address
                inc     hl
                in      a,(c)                   ; read 16bit port
                bit     7,c                     ; test bit 7 (Z = not keyboard, maybe kempston joystick,
                                                high byte is zero)
                jp      z,MOUSE_KB_NCPL
                cpl                             ; keys are active in log. 0
MOUSE_KB_NCPL   and     (hl)                    ; apply bitmask
                inc     hl
                jp      z,MOUSE_KB_CN           ; nothing pressed
                set     5,d                     ; set bit for this key or joystick switch
MOUSE_KB_CN     srl     d                       ; rotate byte
                dec     e                       ; decrement counter
                jp      nz,MOUSE_KB_CLP         ; repeat for all keys
                ret

MOUSE_KEYTAB    db      254,239,8               ; 16b port (2B) and bitmask (1B)
                db      254,239,16
                db      254,239,4
                db      254,239,2
                db      254,239,1

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37750 - rightl, compute address of byte right from first

RIGHTL          ld      a,l
                inc     l
                xor     l
                bit     5,a
                ret     z
                ld      h,0
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37759 - downhl, compute address in VRAM under address of line in HL

DOWNHL          inc     h
                ld      a,h
                and     7
                ret     nz
                ld      a,l
                add     a,32
                ld      l,a
                ld      a,h
                jr      c,DOWNHL2
                sub     8
                ld      h,a
DOWNHL2         cp      88
                ret     c
                ld      h,0
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37780 - clear screen

CLS             ld      hl,16384                ; clear pixels
                ld      de,16385
                ld      bc,6144
                ld      (hl),0
                ldir
                ld      bc,767                  ; clear atributes
                ld      (hl),56                 ; white paper, bright 0, black ink
                ldir
                ret
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
; 37801 - PAUSE 0, wait until key is not pressed

PAUSE0          in      a,(254)
                cpl
                and     31
                ret     nz
                in      a,(31)
                bit     7,a
                jp      nz,PAUSE0
                and     01110000b               ; 3 fire buttons for AMouse or Kempston Joystick
                jp      z,PAUSE0
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37820 - PAUSENK, wait until key is pressed

PAUSENK         in      a,(254)
                cpl
                and     1
                jp      nz,PAUSENK
                in      a,(31)
                and     01110000b               ; 3 fire buttons for AMouse or Kempston Joystick
                jp      nz,PAUSENK
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 37836 - informations about selected file

FILE_INFO       call    SCREEN_BACKUP
                call    FIND_SELFILE
                jp      nz,MOUSE_GUI            ; NZ = file not found, start mouse driver
                push    hl
                ld      hl,FILEINFO_TXT_2
                call    TEXTOUT_1
                call    A_38006
                ld      hl,FILEINFO_TXT_1
                call    TEXTOUT_1
                pop     hl
                ld      b,8
                call    TEXTOUT_3
                ld      a,32
                rst     16
                ld      a,60
                rst     16
                ld      a,(hl)
                inc     hl
                rst     16
                ld      a,62
                rst     16
                ld      a,32
                rst     16
                push    hl                      ; ld ix,hl
                pop     ix
                ld      l,(ix+0)
                ld      h,(ix+1)
                call    NUMOUT_16
                ld      a,44
                rst     16
                ld      l,(ix+2)
                ld      h,(ix+3)
                call    NUMOUT_16
                ld      hl,FILEINFO_TXT_3
                call    TEXTOUT_1
                ld      hl,FILEINFO_TXT_L
                call    TEXTOUT_1
                ld      l,(ix+4)
                ld      h,0
                call    NUMOUT_8
                ld      hl,FILEINFO_TXT_S
                call    TEXTOUT_1
                ld      l,(ix+6)
                ld      h,0
```

```
                call    NUMOUT_8
                ld      hl,FILEINFO_TXT_T
                call    TEXTOUT_1
                ld      l,(ix+5)
                ld      h,0
                call    NUMOUT_8
                call    SOUND_CLICK
                call    PAUSENK
                call    PAUSE0
                call    PAUSENK
                call    SOUND_CLICK
                call    SCREEN_RESTORE
                jp      MOUSE_GUI               ; start mouse driver

; 37975 - file info coordinates and messages

FILEINFO_TXT_1  db      22,15,2,17,5
                db      16,9,19,1+128

FILEINFO_TXT_2  db      17,8+128

FILEINFO_TXT_3  db      22,16,2,17,5+128

FILEINFO_TXT_L  db      "LENG",174

FILEINFO_TXT_S  db      " SEC",174

FILEINFO_TXT_T  db      " TRC",174


; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38006 - draw lines - rectangle around fileinfo?

A_38006         exx
                push    hl
                exx
                ld      bc,14351
                call    8933
                ld      bc,193
                ld      de,65281
                call    9402
                ld      bc,4352
                call    9402
                ld      de,511
                ld      bc,193
                call    9402
                ld      bc,4352
                call    9402
                exx
                pop     hl
                exx
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38049 - ISO ROM installer

ISOROM_INSTALL  call    FIND_SELFILE
                jp      nz,MOUSE_GUI            ; NZ = file not found, start mouse driver
                push    hl
                ld      de,8
                add     hl,de
                ld      a,(hl)
                cp      'C'
                pop     ix
                jp      nz,MOUSE_GUI            ; NZ = file is not bytes, start mouse driver
                ld      a,(ix+13)
                cp      64                      ; check length of file - 16kB or 32kB
                jp      z,ISOROM_LOAD           ; both can be ZX Spectrum ROM (1 or 2 pages)
                cp      128
                jp      z,ISOROM_LOAD
                jp      MOUSE_GUI               ; file have not right size, start mouse driver
```

```
;------------------------------------------------------------------------------
; 38084 - picture viewer - prepare

PICTURE_VIEW    call    SCREEN_BACKUP
                call    FIND_SELFILE
                jp      nz,MOUSE_GUI            ; NZ = file not found, start mouse driver
                push    hl
                ld      de,8                    ; compute address of file extension in HL
                add     hl,de
                ld      a,(hl)                  ; is it bytes?
                cp      'C'
                pop     hl
                jp      nz,MOUSE_GUI            ; NZ = file is not bytes, start mouse driver
                jp      TRD_LOAD_PIC

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38108 - copy picture to buffer

SCREEN_BACKUP   ld      hl,16384
                ld      de,VRAM_BUFFER
                ld      bc,6912
                ldir
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38120 - copy picture from buffer to VRAM

SCREEN_RESTORE  ld      de,16384
                ld      hl,VRAM_BUFFER
                ld      bc,6912
                ldir
                ret

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38132 - test and display loaded picture from buffer

PIC_VIEWER      push    af                      ; store all (really is needed?)
                push    bc
                push    de
                push    hl
                push    ix
                call    SOUND_CLICK
                call    CLS
                ld      hl,UNI_BUFFER           ; HL = set source pointer to start of buffer, here are
                                                ▒▒▒▒▒▒▒▒▒▒loaded binary data
                ld      de,20                   ; pattern can be to 20 bytes from start
                ld      ix,PIC_VIEW_PATT        ; pointer to part of decompressor code
                ld      (PIC_VIEW_IX),ix        ; store pointer to pattern
                ld      a,3                     ; parts counter
                ld      (PIC_VIEW_COUNT),a      ; store counter
PIC_VIEW_1      ld      a,(hl)                  ; get byte from loaded binary file and compare with
part/example
                cp      (ix+0)
                jp      z,PIC_VIEW_2            ; Z = bytes are same
                inc     hl                      ; NZ = increment offset of pattern to binary data
                ld      ix,(PIC_VIEW_IX)        ; renew IX pointer
                dec     de                      ; decrement length limit
                ld      a,e
                or      d
                jp      z,PIC_VIEW_NEXT         ; Z = pattern not found, try next pattern
                jp      PIC_VIEW_1

PIC_VIEW_2      inc     ix                      ; next byte
                inc     hl
                dec     de
                ld      a,e
                or      d
                jp      z,PIC_VIEW_NEXT
                ld      a,(ix+0)
                cp      255
                jp      z,PIC_VIEW_FOUND        ; Z = yes, binary data are runable decompressor
                jp      PIC_VIEW_1
```

```
PIC_VIEW_NEXT   ld      ix,(PIC_VIEW_IX)        ; get old IX value and compute start of next pattern
                ld      de,9
                add     ix,de
                ld      de,20
                ld      (PIC_VIEW_IX),ix        ; store pointer to start of new pattern
                ld      a,(PIC_VIEW_COUNT)      ; decrement pattern counter
                dec     a
                ld      (PIC_VIEW_COUNT),a      ; store counter
                ld      hl,UNI_BUFFER           ; set source pointer to start of buffer
                jp      nz,PIC_VIEW_1
                ld      de,16384                ; copy picture to VRAM = display it
                ld      hl,UNI_BUFFER
PIC_VIEW_LEN    ld      bc,6912                 ; length will changed if file is shorter than 6912 bytes
                ldir
                jp      PIC_VIEW_FINISH

PIC_VIEW_FOUND  ld      ix,0                    ; 38246 ?
                call    UNI_BUFFER              ; call and uncompress picture
PIC_VIEW_FINISH call    PAUSENK
                call    PAUSE0
                call    PAUSENK
                call    SOUND_CLICK
                pop     ix
                pop     hl
                pop     de
                pop     bc
                pop     af
                ret

; this 3 sequences are in start of various compressed pictures (Pressor 3, Pressor 5, Pressor 6 etc...)

PIC_VIEW_PATT   db      0                       ; 38273
                call    82
                dec     sp
                dec     sp
                pop     hl
                add     hl,bc
                db      255

                db      0                       ; 38282
                dec     sp
                dec     sp
                pop     de
                ld      hl,36
                add     hl,de
                db      255

                sub     d                       ; 38291
                rla
                dec     sp
                dec     sp
                pop     hl
                ld      de,123
                db      255

PIC_VIEW_IX     db      129,149
PIC_VIEW_COUNT  db      0

;------------------------------------------------------------------------------
; 38303 - quit

RET2BAS_QUIT    ld      a,'6'                   ; return to basic without load o run anything
                jp      RET2BAS_COMMON          ; just start TRDOS command line

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38308 - load or run selected file

RET2BAS_LOAD    call    FIND_SELFILE
                jp      nz,MOUSE_GUI            ; NZ = file not found, start mouse driver
                ld      a,'9'                   ; change address RANDOMIZE USR VAL "15616" to 15619
RET2BAS_COMMON  ld      (BASLN_RANDUSR),a
```

```
RET2BAS_8255    ld      a,155                   ; set 8255 - all ports will inputs (155) or outputs
(128)
                out     (127),a
                ld      a,255                   ; set BASIC error code -1 = OK
                ld      (23610),a
                ld      hl,15360                ; set ROM font for RST 16 output
                ld      (23606),hl
                ld      bc,65533                ; silence AY
                ld      a,7
                out     (c),a
                ld      b,191
                ld      a,255
                out     (c),a
                ld      b,253
                ld      a,10
                out     (c),a
                pop     af                      ; restore all registers
                exx
                pop     bc
                pop     de
                pop     hl
                exx
                pop     ix
                pop     iy
                pop     bc
                pop     de
                pop     hl
                ei
                im      1                       ; IM1 for BASIC
                ret                             ; return to BASIC

;-------------------------------------------------------------------------------
; 38370 - set 8255 as output, this is usefull for loading Sampletracker snapshots etc...

RET2BAS_8255OUT ld      a,128                   ; init 8255 interface as output - for 3 channel D/A
converter
                ld      (RET2BAS_8255+1),a
                call    SOUND_EFFECT
                ret

;===============================================================================
; 38379
                db      1                       ; 38379         not used?

PRINT_FN_TABP_1 dw      HEAD_POINTERS           ; 38380         9+150*256 = 38409
PRINT_FN_TABP_2 dw      HEAD_POINTERS           ; 38382         9+150*256 = 38409
LISTF_PRNPOS    db      22,12,1                 ; 38384 filename printing position
                db      16,8+128

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; file header for autocopy function

TRD_HEADER      db      "boot    B"             ; 38389
                db      0,0                     ; 38398 start address
                db      0,0                     ; 38400 length (bytes) or length of BASIC without
variables
                db      0                       ; 38402 length in sectors
TRD_HEADER_SEC  db      0                       ; 38403 first sector
TRD_HEADER_TRCK db      0                       ; 38404 first track

ICON_INDEX      db      0                       ; 38405 number of selected icon
FILE_TYPE_TMP   db      0                       ; 38406 file type - value used when testing files in
file type switcher
FILE_TYPE       db      1                       ; 38407 file type - for file filter
TRDOS_SELDRIVE  db      0                       ; 38408 number of selected TRDOS drive


; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38409 - pointers to TRDOS files in directory

HEAD_POINTERS   ds      256                     ; 41,152, 233,154, 233,155, 25,156 ...
                                                ; 38953, 39657, 39913, 39961 ...
```

```
                                                ;
                                                ; ArtSv2_3B
                                                ; BASPCL  C
                                                ; DTPman2 C
                                                ; DTPman2FC
                                                ; ...

HEAD_POINTERS_2 ds      256                     ; ??

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 38921 - first 9 sectors from TRDOS disc = directory + system sector
; 41225 - picture (VRAM) buffer
; 48137 - universal buffer for 16k ROM pages etc...

;==============================================================================
```